Multi-Scale Spatially Weighted Local Histograms in O(1)

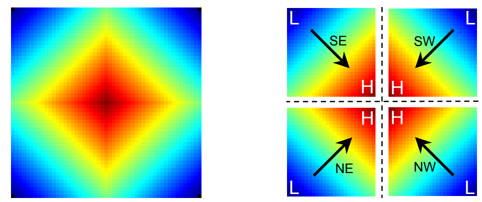
Mahdieh Poostchi, Ali Shafiekhani, Kannappan Palaniappan
Electrical Engineering and Computer Science Department
University of Missouri-Columbia
Columbia, Missouri
Email: {mpoostchi,ashafiekhani,pal}@mail.missouri.edu

Guna Seetharaman
US Naval Research Laboratory
Washington D.C.
Email: guna@ieee.org

Abstract*—Histograms are commonly used to characterize and analyze the region of interest within an image. Weighting the contributions of the pixels to the histogram is a key feature to handle noise and occlusion and increase object localization accuracy of many histogram-based search problems including object detection, tracking and recognition. The integral histogram method provides an optimum and complete solution to compute the *plain* histogram of any rectangular region in constant time. However, the matter of how accurately extract the *weighted* histogram of any arbitrary region within an image using integral histogram has not been addressed. This paper presents a novel fast algorithm to evaluate spatially weighted local histograms at different scale accurately and in constant time using an extension of integral histogram. Utilizing the integral histogram makes it to be fast, multi-scale and flexible to different weighting functions. The pixel-level weighting problem is addressed by decomposing the Manhattan spatial filter and fragmenting the region of interest, subsequently. We evaluated and compared the computational complexity and accuracy of our proposed approach with brute-force implementation and approximation scheme. The proposed method can be integrated into any detection and tracking framework to provide an efficient exhaustive search, improve target localization accuracy and meet the demand of real-time processing.

I. Introduction

In many image processing and pattern recognition applications, sliding window histogram matching is commonly used to detect and localize the target object. Histogram-based features are space efficient, simple to compute, robust to translation and particularly invariant to orientation for color-based features. However, when computing a plain histogram, spatial information are missed which makes it sensitive to noise and occlusion. Several techniques are proposed to preserve spatial information including color Correlograms [1], Spatiogram [2], Multiresolution histogram [3], locality sensitive histogram [4] and fragment-based approaches that exploit the spatial relationships between patches [5]. Spatially weighted histograms boost the performance of many image processing tasks including detection, tracking and recognition at the expense of speed. In [6], Porikli generalized the concept of integral image and presented computationally very fast method to extract the plain histogram of any arbitrary region in constant time. Integral histogram provides an optimum and complete solution for the histogram-based search problem [7], [8], [9], [10], [11]. Since then many novel approaches have



(a) Manhattan (b) 4-Directional Independant Weights

Fig. 1. Illustration of decomposing Manhattan spatial filter into four independent weighting functions. As it is shown in (b), weights linearly increase from one corner to its diagonally opposite corner in each of the quadrants covering four directions : {SE, SW, NE, NW}.

been presented based on integral histogram to accelerate the performance of image processing tasks and incorporate the spatial information including filtering [12], [13], [14], [15], classification and recognition [3], [16], [17], detection and tracking [18], [19], [20].

Despite all different techniques that have been proposed to adaptively weight the contribution of pixels to local histograms, the problem of how accurately extract the spatially weighted histogram of any arbitrary region within an image in constant time using integral histogram is still unsolved. Fragtrack [5] proposes a discrete approximation scheme instead of the continuous kernel weighting approach to give higher weights to the contribution of inner rectangles compare to region margins to meet the demand of real-time processing at the expense of losing accuracy. In this paper, we present a novel fast algorithm to efficiently and accurately evaluate *Spatially Weighted Local Histograms* (*SWLHs*) in O(1) time complexity at multiple scales using an extension of the integral histogram method. The main idea is to

1) Decompose the spatial filter into independent weights w_{i_s} (Figure 1(b)),
2) For all w_{i_s} compute weighted integral histograms of image $IH_{w_{i_s}}$ (second row of Figure 6),
3) Fragment the arbitrary region of interest into multiple quadrants q_{i_s} (following weighting function decomposition fashion, Figure 5),
4) For every quadrant q_i, compute its weighted local histogram using the corresponding IH_{w_i} and considering

978-1-5386-1235-4/17/$31.00 ©2017 IEEE

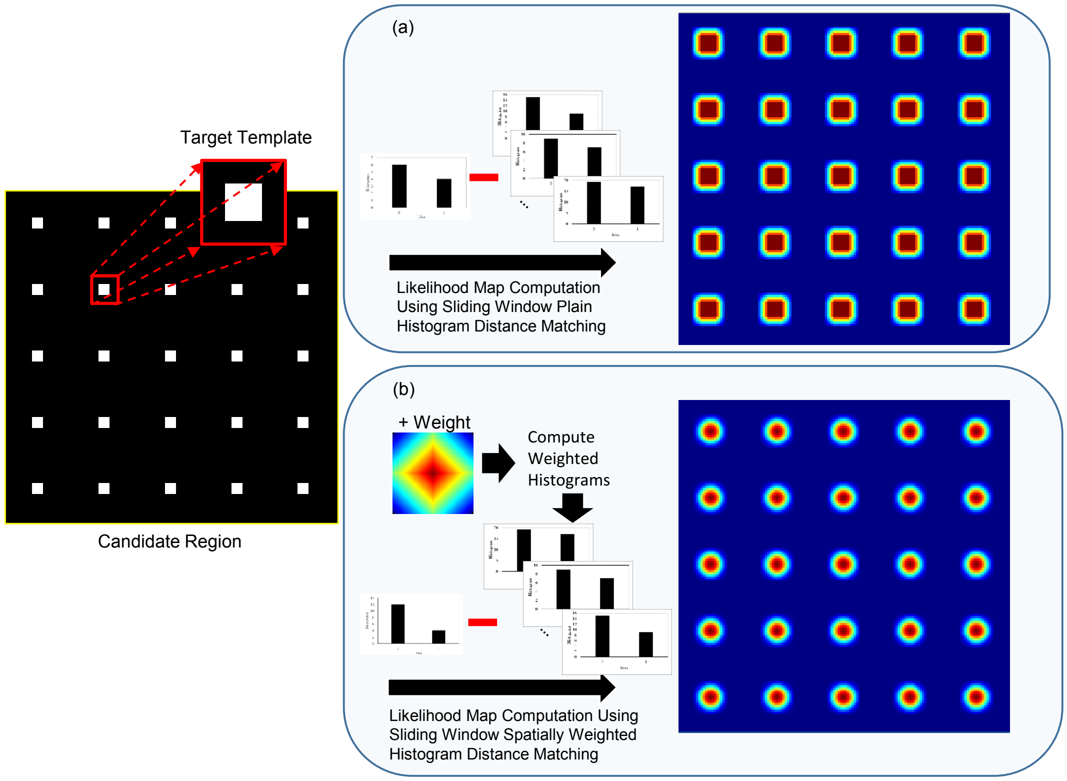


Fig. 2. Performance evaluation of intensity feature likelihood maps using sliding window (b) plain versus (c) spatially weighted histogram distance matching.

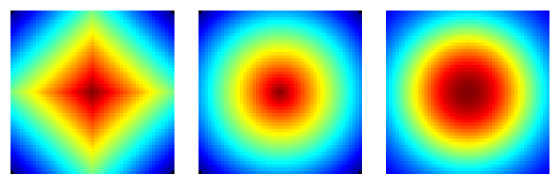
its translation from center pixel,

5) Normalize the local histograms patches,
6) Combine local histograms patches to build the full region of interest weighted local histogram (3rd row of Figure 6)

In the following sections, first we describe the brute-force approach and the approximation scheme [5] to compute the spatially weighted local histograms and then discuss our proposed accurate and fast algorithm (*SWLHs*). Section 3 evaluates and compares the computational complexity and accuracy performance of *SWLHs* with the brute-force implementation and approximation scheme. Section four presents the application of *SWLHs* in context of video object tracking framework.

II. Spatially Weighted Local Histograms

In many object detection and tracking frameworks, sliding window histogram distance matching is commonly used to detect and localize the target object. Figure 2 shows the accuracy of intensity feature likelihood maps based on sliding window histogram matching when using plain local histograms versus spatially weighted local histograms. As it shown in Figure 2(b), weighting pixel contributions is a key feature to increase the accuracy of detection results. The common technique to adaptively weight the contributions of pixels to the histogram is to define a weighting function $w(x, y)$ that assigns weights to pixels with respect to their distance from target center (since undesirable pixels are usually considered around the region contours) including Manhattan, Euclidean, Gaussian or exponential weighting distance functions (Figure 3). Having such kernels enables us to adaptively weight the contributions of pixels and diminish the presence of background information when computing weighted local histograms.



(a) Manhattan (b) Euclidean (c) Gaussian

Fig. 3. Illustration of linear and non-linear distance kernels.

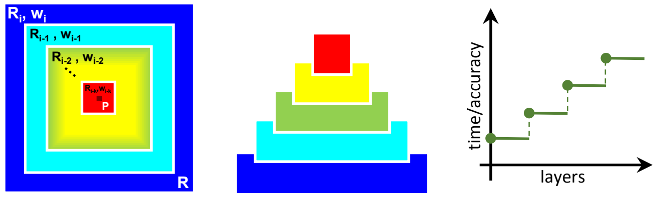
Then the spatially weighted histogram of region T is computed as:

$$H(T, b_i) = \sum_{x,y \in T}^{w \times h} \delta(Q(f(x, y)) - b_i) \times w(x, y) \qquad (1)$$

where T is the region of interest of size $w \times h$, b_i is the histogram bin index, δ is the pulse function and Q is the quantization function for image feature values f.

A. Brute-force Approach

The computational complexity of the straightforward convolution-based approach to compute the adaptively



(a) (b) (c)

Fig. 4. (a) Wedding-Cake Approach: the discrete approximation scheme to compute the spatially weighted local histogram of the candidate region considering inner-nested windows and using integral histogram ($w_i < w_{i-1} < \ldots < w_{i-k}$). (b) Wedding-cake slice approximation. (c) Computational time complexity and accuracy of this method increase by increasing the number of layers.

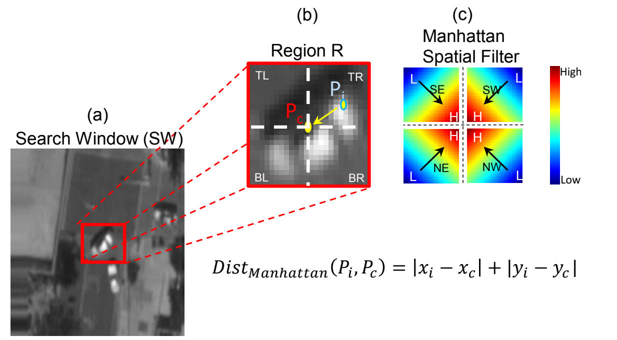


Fig. 5. Decompose the spatial filter into four independent weighting functions w_{i_s}, covering four directions: {SE, SW, NE, NW} and subsequently fragment region of interest into multiple quadrants: TopLeft(TL), TopRight(TR), BottomLeft(BL) and BottomRight(BR).

weighted local histograms at each candidate pixel location is linear to the kernel size and the number of candidate pixels. Assuming a search window of size $w \times h$ and a neighborhood of size $k \times k$ and b-dimensional histogram, the computational complexity of finding the best matched pixel location is $O(b \times k^2 \times w \times h)$, which makes the system far away from real-time performance particularly when it comes to large scale high resolution image analysis.

B. Wedding-Cake Approach

One solution to meet the demands of real-time implementation is to extract local histograms in constant time using integral histogram. However, as of our knowledge, there is no solution to accurately and efficiently compute spatially weighted local histograms in O(1) using integral histogram. Fragtrack [5] proposed a discrete scheme to approximate the kernel function with different weights instead of pixel-level continuous weighting. Assuming that we want to calculate a spatially weighted local histogram in the rectangular region R centered at point P using integral histogram. Such counting can be approximated by considering several inner-nested windows R_i at multiple scales around P (Figure 4(a)). The goal is to compute the counts of the rings between two adjacent windows R_i and R_{i-1} by subtracting their local histograms that are obtained in constant time using integral histogram. Then, rings histograms will be weighted appropriately with respect to their distance from P and combined to approximate spatially weighted local histogram on R as:

$$SWLH_{approximate}(R) = w_i \times (H(R_i) - H(R_{i-1})) + \ldots + w_{i-k} \times H(R_{i-k}) \quad (2)$$

Although the weighted local histogram computations are invariant to kernel size, but the accuracy of this approximation relies on the number of considered inner-nested windows. The computational complexity and accuracy increase by increasing the number of layers.

We present a new algorithm called *SWLHs* to efficiently and accurately compute spatially weighted local histograms in constant time using an extension of integral histograms (Figure 6).

C. Multi-scale Spatially Weighted Local Histograms in O(1) (SWLHs)

In our algorithm, we propose to address the expensive continuous pixel-level weighted local histogram computations using an extension of integral histogram method and Manhattan distance function. *SWLHs* weights the contribution of each pixel $P_i = (x_i, y_i)$- within region R centered at $P_c = (x_c, y_c)$, Figure 5(b) - to the histogram of region R using its Manhattan distance from P_c. Manhattan or city-block weighting function measures the sum of the absolute distance between two points along each axis. In our case, Manhattan distance of any arbitrary point $P_i = (x_i, y_i)$ within region R is

$$Dist_{Manhattan}(P_i, P_c) = |\ x_i - x_c\ | + |\ y_i - y_c\ | \quad (3)$$

Since the filter is rectilinear and symmetric, we propose to decompose it into four independent weighting functions w_{i_s}, covering four directions: {SE, SW, NE, NW} (Fig. 5(c)). These weights are extended to compute four differently weighted integral histogram of the search window (Fig. 6). For each direction, we consider two correlated

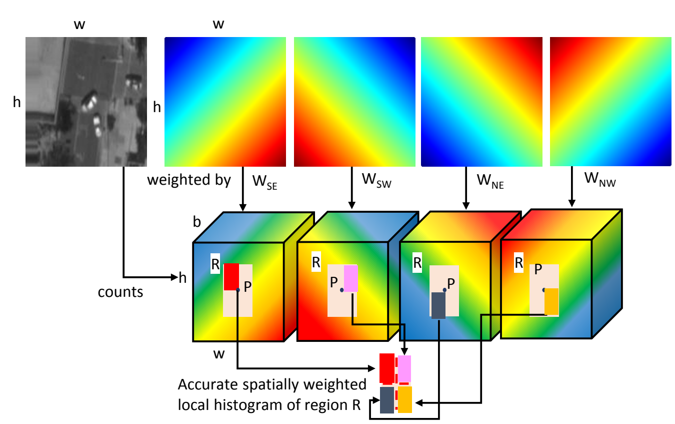


Fig. 6. Compute four weighted integral histogram of search window using $w_{SE}, w_{SW}, w_{NE}, w_{NW}$

images f and w_{dir} to compute the weighted integral histogram up to point (x, y):

$$IH_{w_{dir}}(x, y, b_i) = \sum_{i \leq x, j \leq y} \delta(Q(f(i, j)) - b_i) \times w_{dir}(i, j) \tag{4}$$

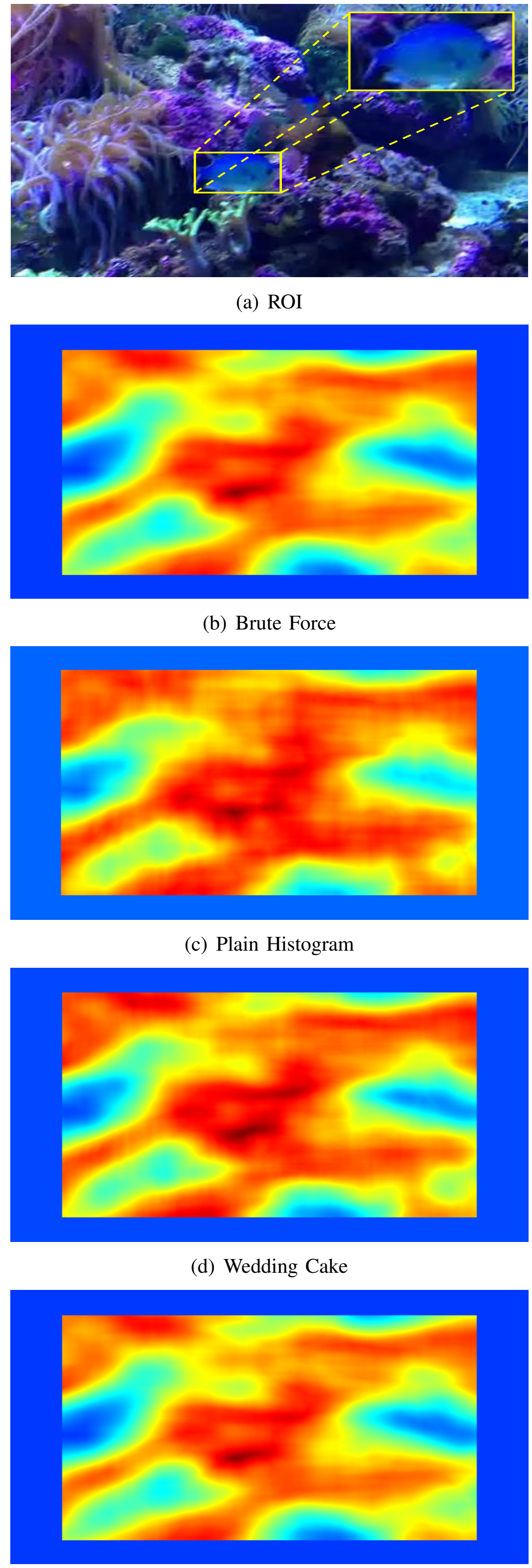
f contains image feature values, Q is the quantization function that determines which bin to increase, δ is the impulse function and w_{dir} is the pixel-wise weighing function that determines the value to increase at that bin.

Now to compute spatially weighted local histogram of any arbitrary region at constant time, first the interested region R is fragmented into four quadrant q_{i_s}: TopLeft(TL), TopRight(TR), BottomLeft(BL) and BottomRight(BR) (Fig. 5(b)). Then the local weighted histogram of each of the quadrants will be computed independently using its corresponding weighted integral histogram and considering its translation from center pixel. Finally, the local histogram of patches are normalized and combined to build the full region spatially weighted histogram (Fig. 6).

It is noteworthy to mention that due to weights rectilinear changes, their values are independent of the pixel location in the region of interests. This characteristic enables us to appropriately normalize the computed weighted local histogram and match it with the target spatially weighted histogram regardless of its location. This new method provides multi-scale accurate spatially weighted local histogram in constant time and can be utilized for other spatial weighting functions. It also can be easily adapted to any fast computations of integral histogram on GPUs to accelerate the computations of four weighted integral histogram [8].

III. Experimental Results and Performance Evaluation

In this section, we evaluate the performance of our approach and compare it with brute-force implementation and approximation scheme with respect to computational complexity and accuracy. Figure 7 illustrates the performance of the estimated intensity likelihood maps for a sample image from the VOT2016 data set [21] using sliding-window histogram matching. We compared the intensity likelihood map computed by the brute-force implementation with the matching results of the plain histogram, approximation scheme and our proposed algorithm *SWLHs*. Background clutter is one of the main challenges in object detection using matching. Therefore we selected an image that contains background clutter to make the matching process very challenging. We calculated the Mean-Squared Error (MSE) between the brute-force result which is our reference model and the two other techniques. The MSE between brute-force and our results (*SWLHs*) is 0 as we expected and 0.12% using the approximation scheme with 3 layers respectively. It is proved that our proposed method not only provides exact results as the brute-force approach but is much faster and invariant to of sliding window size changes. In this experiment, for the image of size 345×460 and sliding window of size 61×91 (Figure 7(a)), $SWLHs$ is 4.5 times faster than brute-force implementation.



(a) ROI

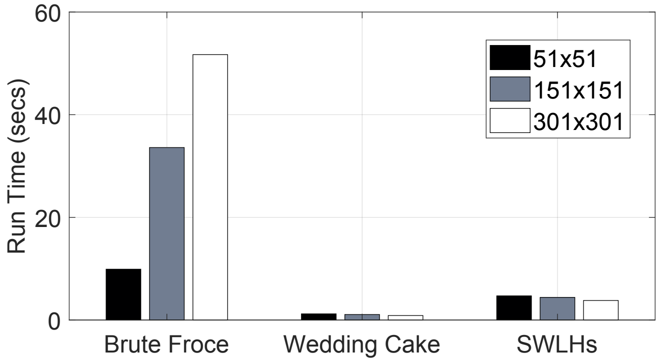
(b) Brute Force

(c) Plain Histogram

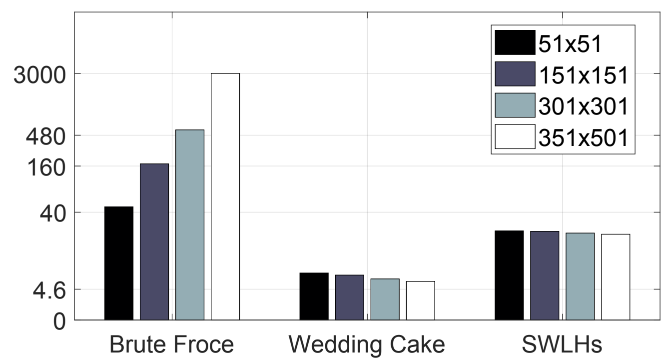
(d) Wedding Cake

(e) SWLHs

Fig. 7. Performance evaluation of intensity likelihood maps estimation using sliding window histogram matching. Weighting pixel contribution considering its location results in more accurate and robust target localization as shown in (b) and (e).



(a) Image Size (640x480)



(b) Image Size (1k x 1k)

Fig. 8. Performance evaluation comparison by increasing the local histogram sliding window size. Computational time complexity of integral histogram-based methods are invariant of kernel size. However, Computational complexity and accuracy of Wedding Cake approach increases by increasing the number of layers

Figure 8 illustrates the computational complexity performance of *SWLHs* compared to brute-force approach and the approximation scheme. Figure 8(a) shows the computational complexity of each of the discussed methods for standard image 640×480 and Fig. 8(b) for large scale image of size $1k \times 1k$, for different sliding window size from small scale to very large scale. It can be seen that the local histograms computational time using the brute-force implementation increases dramatically by enlarging the kernel size but is invariant of sliding window size for the approximation scheme and *SWLHs*. The execution time of the integral histogram based methods are invariant of sliding window size, however there is a small drop in execution time when increasing the sliding window size. The reason is that the number of sliding windows and consequently the number of computed local histograms is reducing by increasing the sliding window size. It is shown that, for a search window of size 640×480 and small kernel of size 51×51 Figure 8(a), $SWLHs$ is two times faster than the brute-force approach and 14 times faster for a larger sliding window of size 301×301. $SWLHs$ is 185 times faster for a large scale image of size $1k \times 1k$ and kernel of size 351×501.

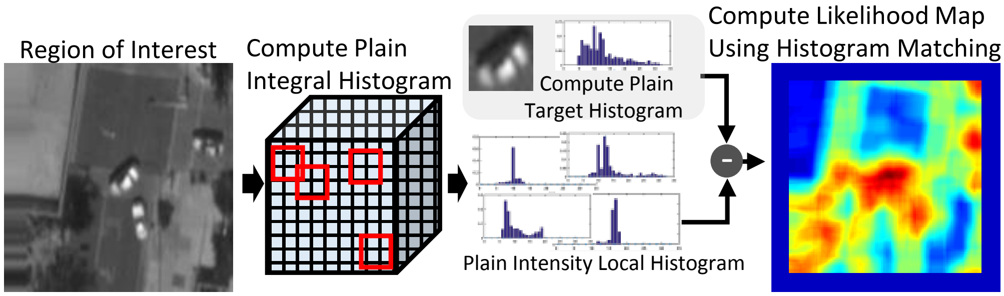
IV. Spatially Weighted Integral Histogram for Fast Tracking (SWIFT)

Many of the discriminative region-based tracking algorithms rely on histograms for a fast and memory efficient appearance modeling of target object as well as candidate regions including Mean-Shift [22] and kernel-based [23], [24] methods. However, since many of these trackers discard the spatial information when computing the conventional histogram, they rapidly lose the accuracy and converge to false targets. Therefore, many techniques have been presented to incorporate spatial information and enhance the tracker robustness which are either more computationally intensive and far away from real-time performance or a combination of different techniques to compensate the lack of spatial information [2], [25], [4], [18], [26]. Our proposed approach that incorporates spatial information to the histogram of color, shape or texture features presents a novel solution based on an extension of integral histogram to efficiently compute histogram-based matching for visual detection, tracking and recognition.

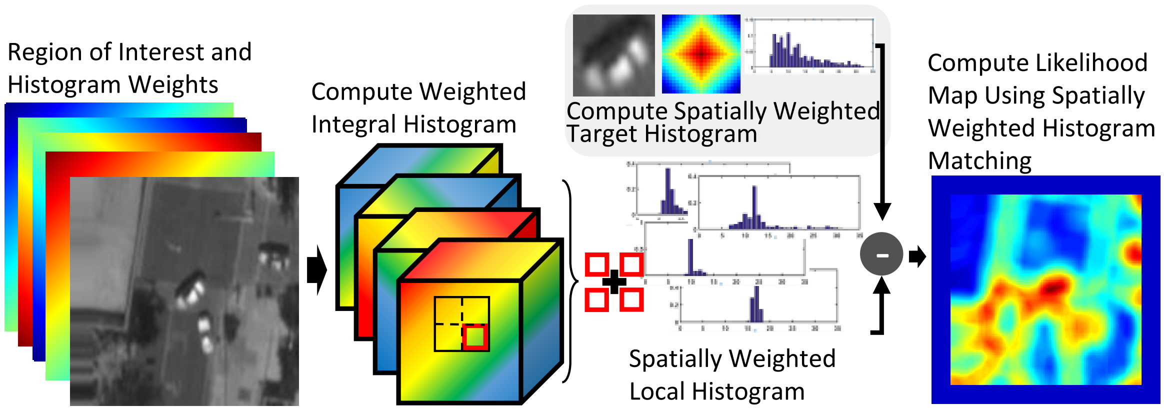
We integrated *SWLHs* into our tracking system named *LoFT* [27], [9] to evaluate the performance of intensity histogram matching when using plain histogram versus spatially weighted histogram. *LoFT* is an appearance-based Likelihood of Features Tracking (LoFT) system, specialized for low resolution targets with large displacements caused by low frame rate sampling in Wide Area Motion Imagery(WAMI). Matching likelihood maps for individual features are computed using sliding window histogram similarity operators. The integral histogram method is used to accelerate computation of the sliding window histograms for a posteriori likelihood estimation [8]. Figure 9(a) describes the flow of the likelihood map computation using plain intensity histogram of target object and candidate regions. Similar to the results obtained for the synthetic image shown in Figure 2, using plain histograms results in less accurate localization of object. We applied our accurate spatially weighted integral histogram to estimate features likelihood maps instead of regular integral histogram to perform fast exhaustive search. Figure 9(b) illustrates the computational flow compared to plain histogram and presents the more accurate target localization results when using spatially weighted histograms.

V. Conclusion

This paper presents our novel fast algorithm to accurately evaluate spatially weighted local histograms in constant time using an extension of the integral histogram method ($SWLHs$). We have shown that $SWLHs$ computes accurately spatially weighted local histograms compared to brute-force approach and meets the demands of real-time processing. Utilizing the integral histogram makes it to be fast, multi-scale and flexible to different weighting functions. This technique can be applied to fragment-based approaches to adaptively weight object patches considering their location. $SWLHs$ can be integrated into any detection and tracking framework to provide an efficient exhaustive search and achieve more robust and accurate target localization.



(a) Likelihood map estimation using sliding window plain histogram matching



(b) Likelihood map estimation using sliding window spatially weighted histogram matching

Fig. 9. Performance evaluation of intensity likelihood map estimation using sliding window (a) plain histogram versus (b) our proposed accurate spatially weighted histogram matching approach to perform presistant tracking of moving vehicles in large scale aerial imagery.

VI. Acknowledgment

This research was partially supported by the U.S. Air Force Research Laboratory under agreement AFRL FA875014-2-0072. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of AFRL, NRL, or the U.S. Government.

References

[1] Jing Huang, S Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih, "Image indexing using color correlograms," in *IEEE CVPR*, 1997, pp. 762–768.

[2] Stanley T Birchfield and Sriram Rangarajan, "Spatiograms versus histograms for region-based tracking," in *IEEE CVPR*, 2005, vol. 2, pp. 1158–1163.

[3] Efstathios Hadjidemetriou, Michael D Grossberg, and Shree K Nayar, "Multiresolution histograms and their use for recognition," *IEEE PAMI*, vol. 26, no. 7, pp. 831–847, 2004.

[4] Shengfeng He, Qingxiong Yang, Rynson W.H. Lau, Jiang Wang, and Ming-Hsuan Yang, "Visual tracking via locality sensitive histograms," in *IEEE CVPR*, 2013, pp. 2427–2434.

[5] Amit Adam, Ehud Rivlin, and Ilan Shimshoni, "Robust fragments-based tracking using the integral histogram," in *IEEE CVPR*, 2006, vol. 1, pp. 798–805.

[6] Fatih Porikli, "Integral histogram: A fast way to extract histograms in cartesian spaces," in *IEEE CVPR*, 2005, vol. 1, pp. 829–836.

[7] Mahdieh Poostchi, Kannappan Palaniappan, Filiz Bunyak, and Guna Seetharaman, "Realtime motion detection based on the spatio-temporal median filter using GPU integral histograms," in *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*. ACM, 2012, p. 19.

[8] Mahdieh Poostchi, Kannappan Palaniappan, Filiz Bunyak, Michela Becchi, and Guna Seetharaman, "Efficient GPU implementation of the integral histogram," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 266–278.

[9] Mahdieh Poostchi, Filiz Bunyak, Kannappan Palaniappan, and Guna Seetharaman, "Feature selection for appearance-based vehicle tracking in geospatial video," in *Geospatial InfoFusion III*. International Society for Optics and Photonics, 2013, vol. 8747, p. 87470G.

[10] Mahdieh Poostchi, Kannappan Palaniappan, and Guna Seetharaman, "Spatial pyramid context-aware moving vehicle detection and tracking in urban aerial imagery," in *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017, pp. 1–6.

[11] Kannappan Palaniappan, Mahdieh Poostchi, Hadi Aliakbarpour, Raphael Viguier, Joshua Fraser, Filiz Bunyak, Arslan Basharat, Steve Suddarth, Erik Blasch, and Raghuveer M Rao, "Moving object detection for vehicle tracking in wide area motion imagery using 4D filtering," in *International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 2830–2835.

[12] Fatih Porikli, "Constant time o (1) bilateral filtering," in *IEEE CVPR*, 2008, pp. 1–8.

[13] Ke Zhang, Gauthier Lafruit, Rudy Lauwereins, and Luc Van Gool, "Joint integral histograms and its application in stereo matching," in *IEEE ICIP*, 2010, pp. 817–820.

[14] Yonghuai Liu, Luigi De Dominicis, Ralph R Martin, and Baogang Wei, "Illumination normalization using weighted gradient integral images," *Lecture Notes on Software Engineering*, vol. 1, no. 4, pp. 329, 2013.

[15] Mahdieh Poostchi, Hadi Aliakbarpour, Raphael Viguier, Filiz Bunyak, Kannappan Palaniappan, and Guna Seetharaman, "Semantic depth map fusion for moving vehicle detection in aerial video," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 32–40.

[16] Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto, "Localizing objects with smart dictionaries," in *European Conference on Computer Vision*, 2008, pp. 179–192.

[17] Ali Shafiekhani, Suhas Kadam, Felix B Fritschi, and Guilherme N DeSouza, "Vinobot and Vinoculer: Two robotic platforms for high-throughput field phenotyping," *Journal of Sensors*, vol. 17, no. 1, pp. 214, 2017.

[18] Qi Zhao and Hai Tao, "Object tracking using color correlogram," in *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005, pp. 263–270.

[19] Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge Belongie, "Integral channel features," *BMVC Press*, 2009.

[20] Ali Shafiekhani, Mohammad J Mahjoob, and Mehdi Akraminia, "Design and implementation of an adaptive critic-based neuro-fuzzy controller on an unmanned bicycle," *Journal of Mechatronics*, vol. 28, pp. 115–123, 2015.

[21] *VOT 2016 Benchmark Data*, http://www.votchallenge.net/vot2016/dataset.html.

[22] Dejun Wang, Weiping Sun, Shengsheng Yu, Lin Li, and Wei Liu, "A novel background-weighted histogram scheme based on foreground saliency for mean-shift tracking," *Multimedia Tools and Applications*, pp. 1–19, 2015.

[23] Jaideep Jeyakar, R Venkatesh Babu, and KR Ramakrishnan, "Robust object tracking with background-weighted local kernels," *Computer Vision and Image Understanding*, vol. 112, no. 3, pp. 296–309, 2008.

[24] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer, "Kernel-based object tracking," *IEEE PAMI*, vol. 25, no. 5, pp. 564–577, 2003.

[25] Wangsheng Yu, Xiaohua Tian, Zhiqiang Hou, Anqi Huang, and Xiang Liu, "Region edge histogram: A new feature for region-based tracking," in *IEEE 12th International Conference on Signal Processing (ICSP)*, 2014, pp. 1180–1185.

[26] Jwu-Sheng Hu, Chung-Wei Juan, and Jyun-Ji Wang, "A spatial-color mean-shift object tracking algorithm with scale and orientation estimation," *Pattern Recognition Letters*, vol. 29, no. 16, pp. 2165–2173, 2008.

[27] R. Pelapur, S. Candemir, F. Bunyak, M. Poostchi, G. Seetharaman, and K. Palaniappan, "Persistent target tracking using likelihood fusion in wide-area and full motion video sequences," in *15th Int. Conf. Information Fusion*, 2012, pp. 2420–2427.